

Understanding AWS in 90 Minutes



Applying the Shared Responsibility Model and Well-Architected Framework

Instructor: Bill Boulden

Today's Agenda

- About Me, About You
- The Well Architected Framework (15 minutes)
- The Shared Responsibility Model (10 minutes)
- The Acceptable Use Policy (5 Minutes)
- Q&A (5 Minutes)
- Quick Hits: Every AWS Service in 20 Minutes
- Break (5 Minutes)
- Hands On Exercise: Walking Through the EC2 Wizard (25 Minutes)

About Me, About You

Hi! I'm Bill "Downupright" Boulden!

- Been developing since age six.
- Twenty-two years real-world programming experience, moving through Basic to VB to .NET to Perl to Node.js.
- Fractional CTO of four (!) tech startups: Use Kitch, Hello Audio, Connector Street, and Adlibz.
- Mix of professional trainer, AWS consultant, and professional musician.
- AWS Certified Developer
- Hobbies include producing and DJing Electronic Dance Music.
- twitter.com/Downupright
- linkedin.com/in/Spruke
- They/them pls :)



About You

- Prerequisites: ALMOST. NONE.
- I have done my utmost to make this course simple enough that it can be anyone's "My First AWS Course".
- You need to have an AWS account ready to go if you wish to follow along with the live walkthrough exercise.
- You need to be literate in basic networking concepts such as what an IP address is and what DNS does.
- If and only if you wish to play with the virtual server you create during the walkthrough, you need to be able to use a Terminal application, preferably PuTTY on Windows or the built-in Terminal on Mac.
 - Today I will be using PuTTY in the training.

The Well Architected Framework

What is the Well Architected Framework?

- A set of guidelines for how AWS believes you should use AWS- the official “Best Practices” theory.
- While there’s More Than One Way To Do Anything, there is also a Best Way.
- It is recommended that you undergo a Well-Architected Review, where an Amazon Premier Partner will evaluate you on each of the five pillars and make a series of recommendations on how to improve your adherence to the WAF.
- You can receive credits for undergoing a Well-Architected Review. In the past, I’ve received up to \$5,000 in credits.
- I’ve done two at my various startups and am preparing to undergo a third!

Pillar: Operational Excellence

- Formal definition:
 - The Operational Excellence pillar includes the ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value.
- Translation:
 - Strong change management and DevOps.
- Sample WAR question:
 - “How do you mitigate deployment risks?”

Keys to Operational Excellence

- Infrastructure As Code
 - This specifically means CloudFormation!
- Formal & documented Change Management Processes
- Safe experimentation in production-like environments
- Rapid rollbacks
- Observability tools, insight into what's happening under the hood
 - CloudWatch, X-Ray, CloudTrail

Pillar: Security

- Formal definition:
 - “The security pillar describes how to take advantage of cloud technologies to protect data, systems, and assets in a way that can improve your security posture.”
- Translation:
 - Secure your AWS environment and secure the applications you build in it
- Sample WAR question:
 - “How do you manage identities for people and machines?”

Keys to Security

- Principle of least privilege access necessary
- IAM (Identity Access & Management) users, roles, groups, policies
- Encryption of data at rest & in transit
 - Often as simple as a single checkbox on many major services such as S3, RDS, DynamoDB
- Enforcing MFA, particularly on the root account
- Separation of root account from “active” accounts
 - The Root Account is the one that you log into with email and password. It has access to billing and the ability to fundamentally alter or delete the account. It has super access to everything. It should be used extremely infrequently and only for billing operations.
 - IAM accounts are the ones you log into with username and password. They should be used for all action-taking operations such as creating and modifying resources, with their access policies defined downward to the minimum necessary.

Pillar: Reliability

- Formal definition:
 - “The Reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it’s expected to. This includes the ability to operate and test the workload through its total lifecycle.”
- Translation:
 - Fault tolerance, redundancy, uptime
- Sample WAR question:
 - “How do you back up data?”

Keys to Reliability

- For serverful architectures:
 - Distributing workloads across Regions and Availability Zones for redundancy
 - Use of Auto Scaling Groups with policies that automatically replace servers
- For serverless architectures:
 - Dead-letter-queues, automated retries
- For all architectures:
 - Observability for failures (Cloudwatch)
 - Use of S3 with its 11 9's of durability
 - Monitoring service quotas
 - Horizontal Scaling

Pillar: Performance Efficiency

- Formal definition:
 - “The Performance Efficiency pillar includes the ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.”
- Translation:
 - Use the right services for the job, and use the right amount of them.
- Sample WAR question:
 - “How do you select your database solution?”

Keys to Performance Efficiency

- Preferring serverless to serverful
- If AWS offers “XYZ as a service”, it is nearly always correct to use that rather than use general purpose computing (EC2/containers) to accomplish the same goal
 - Example: If you have need of a Postgres database, it is correct to use RDS (database-as-a-service) to create one rather than creating an EC2 and installing Postgres software on it.
- Using S3 for storage
- Reviewing new AWS services as they are released to see if a new “XYZ as a service” replaces something you were doing manually

Pillar: Cost Optimization

- Formal definition:
 - “The Cost Optimization pillar includes the ability to run systems to deliver business value at the lowest price point.”
- Translation:
 - AWS has a reputation for being expensive if you don’t use it wisely. Contrary to what you think, AWS does not want you to pay it lots of money. It wants to counteract that reputation by providing all your needs at the lowest price point possible.
- Sample WAR question:
 - “How do you monitor usage and cost?”

Keys to Cost Optimization

- Overlap with Performance Efficiency—using the appropriate “XYZ as a service” is cheaper than general purpose compute
- On-demand vs. Reserved vs. Spot Instances
- Use of Budgets in the Cost Explorer
- S3 Storage Classes
- Decommissioning resources when they are done being used, turning them off during off-hours

The Shared Responsibility Model

Shared Responsibility Model Basics

- A breakdown of what is the CUSTOMER'S responsibility vs. AWS'S responsibility
- AWS does Security OF the Cloud
- Customer does Security IN the Cloud
- Formal definition:
 - “Security and Compliance is a shared responsibility between AWS and the customer. This model can help relieve the customer's operational burden as AWS operates, manages and controls the components from the host operating system and virtualization layer down to the physical security of the facilities... The customer assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software as well as the configuration of the AWS provided security group firewall.”
- Translation: If something goes wrong, is it AWS's fault or yours?

AWS: Physical & Personnel Security at the Datacenter

- Securing the building
- Securing the servers
- Environmental controls
- Best practices for keycard access, password rotation etc. among the employees at the datacenter
- For various ISO certifications, HIPAA/Sarbox compliance, they cover everything related to password policies

More AWS Responsibilities

- Ensuring complete logical separation of network traffic and data that belongs to different clients even when it's sharing hardware and pipes
- Enforcing the firewalls that the customer creates
- Enforcing the encryption that the customer has specified

Customer Responsibilities

- Securing the individual compute resources created
 - If you make an EC2, it's on you to keep the OS up to date and patched; it's also on you to keep all libraries up to date and patched.
- Protecting customer data
 - If you make an RDS and check encryption, then it's on AWS to make sure your data is encrypted... but on you to make sure that you didn't leave your application open to 'SELECT * FROM USERS;'
- Accurately specifying the protections you wish AWS to enforce.
 - If a bad packet comes in on port 999 and crashes your server, and you never specified a firewall, that's your fault. If you specified a firewall that should block port 999 and yet somehow that packet made it through anyway, that would hypothetically be AWS's fault (though in practice that never happens).

The Acceptable Use Policy

What You Can't Do on AWS - Bad Actors

- Distribute viruses or malware
- Send spam
 - More generally, send mass email at all.
 - SES is for transactional email like per-account notifications.
 - EC2s and Lambdas override individual firewall settings to block port 25
- Attempt to phish or pharm

What You Can't Do on AWS - Illegal Stuff

- Pornography
 - Strictly legal pornography is acceptable, but pornographic companies that operate at scale have historically had difficulty verifying all material for consent/age/local law violations. It is safest to not use AWS for pornography at all.
- Copyrighted Material
- Offer a service that is Fraudulent in nature such as a ponzi scheme or product that is known to fail
- Hateful Content

Gray Area: Mine Crypto

- It is not strictly a violation of the Acceptable Use Policy, but they frown on it.
- You will not receive support and word-of-mouth has it your systems will regularly be flagged or erroneously shut off for hogging all the resources.
- **YOU WILL LOSE MONEY.**
 - If the current price of Bitcoin/Ethereum/etc was such that it turned a profit on the returned coins after the expense of using AWS compute resources, the price point would adjust immediately such that this was not the case.
- Don't do it.

Only With Permission: Penetration Testing

- It is okay to run penetration tests, stress tests, or white-hat hacking against your own systems on AWS if you **explicitly write ahead and get permission**.
- It is a violation of the AUP to perform these activities without notifying AWS in advance.
- If you attempt to penetration test, stress tests, or white-hat hack your own infrastructure without giving them advance notice, they will pre-emptively notice that your client ID is the source of lots of bad traffic and deactivate resources of yours for safety.

Q&A (5 Minutes)

Buckle Up: Every* AWS Service in 20 Minutes



* - almost every. There are hundreds and even I don't know them all.

Virtual Servers

- EC2- the classic, the original. Virtual servers on shared hardware similar to VMWare. General purpose computing.
 - Associated services: Auto Scaling Groups, Elastic Load Balancers
- Container services:
 - ECS - Elastic Container Service
 - EKS - Elastic Kubernetes Service
 - Fargate - Serverless Containers
- Lightsail - actually EC2's but simple as heck

Code As A Service

- Lambda - executes individual functions (in a variety of programming languages) on an ad-hoc on-demand basis and you pay by the millisecond of elapsed compute time
 - Go, Rust, Java, .NET, Python, Node, Ruby, or custom runtimes
- The glue that holds AWS together
 - Dozens of other services have “hooks” such that “when XYZ event occurs, run this Lambda”
 - Little code snippets can augment your S3 buckets, Cognito user pools, SQS queues, and so many more

Bundling Services That Deploy Applications With Compute Under The Hood

- Elastic Beanstalk - upload application packages and it provisions Elastic Load Balanced, Auto Scaled EC2 groups to serve them
- App Runner - similar to Elastic Beanstalk, but simpler
- OpsWorks - a puppet/chef approach
- Some CodeStar templates

Developer Tools

- Cloud9 - A web-based IDE similar to Visual Studio Code, hosted on EC2s
- CodeCommit - Distributed source version control, analogous to Github/Bitbucket
- CodeBuild - Automated builds when commits arrive in CodeCommit
- CodeArtifact - Saved generated assets when Codebuilds complete
- CodeDeploy - Automated deploys when CodeArtifacts are available
- CodePipeline - Marries a CodeCommit, CodeBuild, CodeArtifact, and CodeDeploy together into one seamless pipeline that creates an entire CI/CD system analogous to Circle, Travis, Jenkins
- CodeStar - Marries all six of the above into an all-in-one factory where you edit the source code in the browser in Cloud9 and it deploys as you go

Inter-Service Messaging

- SNS - *push*-style notifications that connect any number of Message Producers into a Topic that are pushed to Subscribers
- SQS - *pull*-style notifications that connect any number of Message Producers into a Queue that is then polled by consuming processes for the front N items
 - Includes Visibility Timeout and Long Polling
- SES - transactional email service

Identity Management

- IAM - Identities internal to your AWS account
 - Users
 - Groups
 - Roles (both machines and people)
 - Policies
 - Deny-first except in case of role assumption, use principle of least privilege
- Cognito
 - Identity management for Applications, providing secure username and password management, single-sign-on, social sign-ons. Analogous to Auth0 or Okta

Networking

- VPC - a Virtual Private Cloud. A private CIDR (IP address range) (e.g. 10.10.*.*) that you set aside for your devices to span, within which network traffic belongs to you.
 - Every account has a VPC by default, even if you don't remember configuring one!
- Subnets - specific subsets of VPCs that live inside a single availability zone
- Network ACLs - Access Control Lists for the traffic inside VPCs
- Route Tables - Directs traffic to specified IP ranges to specific network hardware such as Internet Gateways and NAT Gateways
- Internet Gateway - Enables two-way traffic in and out of a VPC with public IPs
- NAT Gateway - Enables one-way traffic out of a VPC without exposing the devices whose traffic is being routed through it

Networking, Continued

- Cloudfront - Global CDN similar to Cloudflare. Provides “edge locations” in most major metro areas where content is cached to be closer to client requests. Can do lots of cool things with Lambdas.
- Route 53 - DNS service. Can register and buy domains as well as administer their nameserving. Similar to GoDaddy.

Storage

- S3 - Simple Storage Service. Technically a key/value store, but where the values are typically files (can be any data). Puts “objects” in “buckets”. Can be connected to Lambdas to do cool things. Several storage tiers available.
 - Standard, Infrequent Access, Glacier, Intelligent Tiering...
- Snowball - hardware for transferring up to 80 TB of data in one chunk from your location to an S3 bucket, through physical drives
- Snowmobile - armored truck of snowballs for moving petabytes of data to the cloud

Databases

- RDS - Traditional Relational Databases as a Service
 - Postgres, MySQL, MSSQL, Oracle, MariaDB, and the proprietary Aurora
- ElastiCache - Redis & Memcached as a Service
- DynamoDB - Document Database / Advanced Key-Value database (scalable)
 - Can do cool things with Lambdas
 - Puts “Items” (rows) that are secretly JSON objects into “Tables”
 - Indexing is very unlike traditional databases and takes some getting used to
- Neptune - Graph DB
- QLDB - Immutable chronological ledger database similar to a blockchain

Artificial Intelligence

- SageMaker - rapidly growing general purpose AI service for machine learning with spinoffs of Jupyter Notebooks and lots of common use cases. Bring your dataset and it can do almost all the rest
- Polly - Text to speech
- Transcribe - Speech to text
- Lex - Conversational chatbots (Lambdas under the hood) (powers Alexa)
- CodeGuru - Code reviews based on AI (?!)
- Comprehend - Sentiment analysis and extraction of meaning from text
- Rekognition - Controversial - extracts textual descriptions of the contents of images and videos - yes, including personal facial recognition 😞

Auditing and Observability

- CloudTrail - logs every change made to an AWS service or configuration with IAM identification of how the change came to be
- CloudWatch - logs for metrics across all systems. Everything that happens in any AWS service generates some form of Cloudwatch log in a Log Group that can be aggregated up to metrics. Similar to splunk/loggly/datadog/sumologic.
- X-Ray - Logs that carry across interconnected AWS systems. Understand the time breakdown of a single request as it journeys through SNS queues, API Gateways, Elastic Load Balancers, Lambdas, and more.

Miscellaneous Cool Stuff

- Ground Station - satellite control as a service
- GameLift - runs video gaming servers
- AppSync - offline GraphQL as a service (similar to Apollo)
- Well-Architected Tool - purports to do WARs for you
- Robomaker - AI for robotics development
- Sumerian - AR & VR as a service
- Connect- navigable phone voice trees (combines Lex, Polly, Lambda, Transcribe)

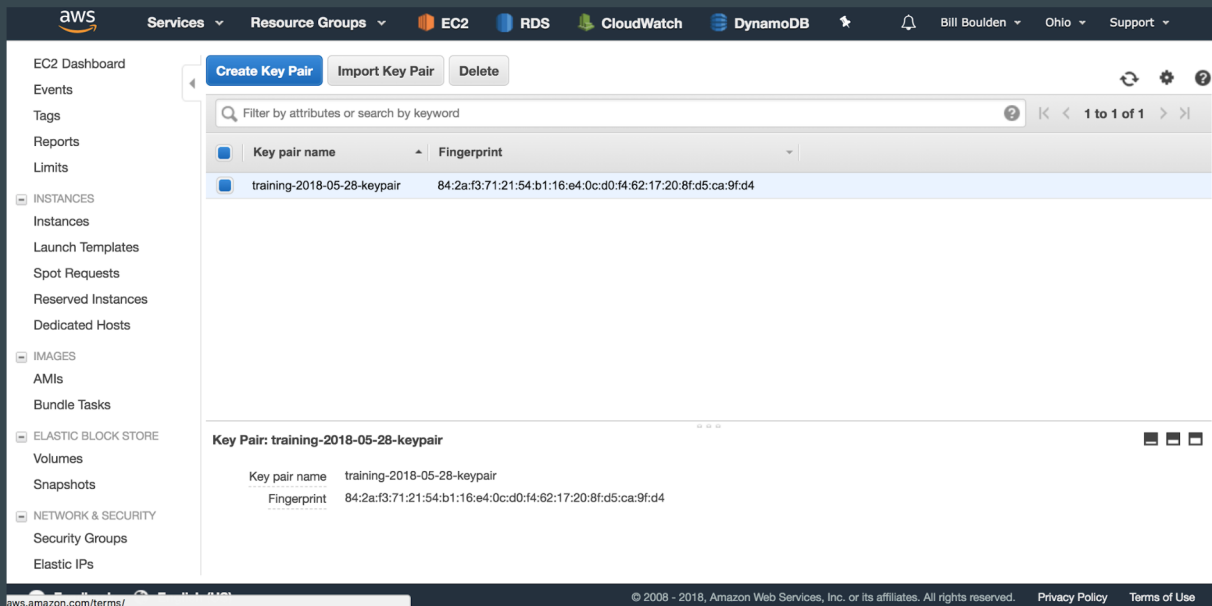
5 Minute Break To Catch My Breath

Spinning Up A Basic EC2 (Interactive Walkthrough)

What's an EC2?

- “Elastic Compute Cloud” aka E.C.C, became EC2 in parlance.
- An EC2 is a virtual server similar to a VMWare instance or OS installation that exists on shared hardware somewhere in an Amazon datacenter but behaves like it is its own distinct server.
- EC2's can be “spun up” from hundreds of different Images.
 - Amazon Linux (a unique Linux distribution most similar to Cent)
 - Ubuntu
 - Windows
 - Special-purpose machines such as premade machines for running logging, firewalls, crypto, deep learning processes, and more!

Generate an SSH Key Pair



The screenshot displays the AWS Management Console interface for the 'Key Pairs' section. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and various AWS services like EC2, RDS, CloudWatch, and DynamoDB. The left sidebar shows the navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area has buttons for 'Create Key Pair', 'Import Key Pair', and 'Delete'. Below these is a search bar and a table of key pairs. One key pair, 'training-2018-05-28-keypair', is listed with its fingerprint. Below the table, the details for the selected key pair are shown.

Key pair name	Fingerprint
training-2018-05-28-keypair	84:2a:f3:71:21:54:b1:16:e4:0c:d0:f4:62:17:20:8f:d5:ca:9f:d4

Key Pair: training-2018-05-28-keypair

Key pair name	training-2018-05-28-keypair
Fingerprint	84:2a:f3:71:21:54:b1:16:e4:0c:d0:f4:62:17:20:8f:d5:ca:9f:d4

aws.amazon.com/terms/

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

How to Prepare the Private Key for Use

- Download and save the .pem file (*this is your only chance to do so!*)
- On a Mac
 - You will need to chmod 600 the .pem file.
 - Then pass it to your terminal ssh command using -i (i for identification) when using ssh.
- On Windows using Putty
 - Use the Puttygen tool to map the .pem into a .ppk.
 - Specify it when connecting with Putty under Connection->SSH->Auth

Launch an EC2 Instance Using The Wizard

- Choose the most recent Amazon Linux AMI (Amazon Machine Image).
- Choose a t2.micro as this is all we will need for today.
- Now let's review the different compute classes.
 - T - burstable CPU
 - M - Medium (well balanced)
 - R - biased toward having more RAM than CPU
 - C - biased toward having more CPU than RAM
 - I - biased towards having more IOPS
 - P - hardware accelerated for TensorFlow (machine learning)
 - G - has onboard graphics cards

Launch an EC2 Instance Using The Wizard, cont.

aws

Services

Resource Groups

EC2

RDS

CloudWatch

DynamoDB

Bill Boulden

Ohio

Support

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances

1

Launch into Auto Scaling Group

Purchasing option

☐ Request Spot instances

Network

vpc-8babe7e3 | training-2018-05-28

Create new VPC

Subnet

subnet-74fc831c | training-2018-05-28-public-us-ea

Create new subnet

Auto-assign Public IP

Use subnet setting (Disable)

Placement group

☐ Add instance to placement group.

IAM role

None

Create new IAM role

Shutdown behavior

Stop

Enable termination protection

☐ Protect against accidental termination

Cancel

Previous

Review and Launch

Next: Add Storage

aws.amazon.com/terms/

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Launch an EC2 Instance Using The Wizard, cont.

- Default Storage options in step four are acceptable as-is.
- In step five, Add Tags, add a tag called “Name” for identification.
- In step six, Security Group, create a new Security Group:
 - With a descriptive name for easy identification.
 - That allows SSH (22) from anywhere.

Launch an EC2 Instance Using The Wizard, cont.

aws

Services ▾

Resource Groups ▾

EC2

RDS

CloudWatch

DynamoDB

★

🔔

Bill Boulden ▾

Ohio ▾

Support ▾

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below.

[Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

Description:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
SSH ▾	TCP	22	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop	✕
Custom TCP I ▾	TCP	80	Anywhere ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕

⚠

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel

Previous

Review and Launch

Launch, Wait 2-3 minutes, Access!

- The user will be “ec2-user”:
 - In PuTTY, enter this under Connection->Data.
 - On Mac, use the command “ssh -i (key).pem ec2-user@(ip-address).”
- Upon access, immediately run “sudo yum update”.

Troubleshooting Break

Parting Q&A

- Follow me on Twitter! twitter.com/downupright
- Connect with me on LinkedIn! linkedin.com/in/spruke
- usekitch.com
- helloaudio.fm
- connectorstreet.com
- Adlibz coming soon